
Week 0, continued

This is CS50. Harvard University. Fall 2014.

Cheng Gong

Table of Contents

| | |
|----------------------------|---|
| Introduction | 1 |
| 3D Printing | 1 |
| Algorithms | 2 |
| Programming Concepts | 3 |
| PB&J Demonstration | 3 |
| Scratch | 5 |
| Announcements | 7 |

Introduction

- [CS50 Staff 2014](#)¹ share favorite memories of CS50, advice for students, and their signature dance moves.
- [Register](#)² for Puzzle Day 2014 with friends from Facebook who will challenge you with problems they have written.

3D Printing

- Ansel Duff, former student and advisee, is now an engineering student who built the binary bulbs this summer with Dan Bradley, as well as the lecturn used on stage.
 - # He built an iPhone dock using a 3D printer, which takes plastic filament and melts it down to build objects layer by layer. To do this, he first measured the device and used CAD (computer aided-design) software to create it.
- On campus, more advanced researchers are using biological materials to solve physiological problems for humans.

¹ <https://www.youtube.com/watch?v=FEUyu607RBE&list=UUcabW7890RKJzL968QWEyKA>

² <http://cs50.harvard.edu/register>

Algorithms

- The field of computer science has many directions and areas to go into. The [guide](#)³ has descriptions and a diagram of classes after 50.
- Today we focus on the fundamental view of having algorithms that find the outputs for a problem given some inputs. The phonebook example was solved in three ways, with one (dividing the problem in half each time) being fundamentally faster. If the phonebook had 4 billion pages, it would only take 32 divisions to find the page containing a particular person like Mike Smith.
- Pseudocode is expressed using English or any language to convey the idea of an algorithm, with the goal of anticipating all possible cases.

```
pick up phone book
open to middle of phone book
look at names
if "Smith" is among names
    call Mike
else if "Smith" is earlier in book
    open to middle of left half of book
    go to line 3
else if "Smith" is later in book
    open to middle of right half of book
    go to line 3
else
    give up
```

The last `else` is a corner case, to handle the possibility that Mike isn't in the phonebook at all.

- Software crashing usually means that a programmer didn't anticipate a case or mistake, so the program acts unpredictably, crashing or rebooting your computer.
- More general problem of taking attendance can be formalized as shown in TED's animated clip "[What's an algorithm?](#)⁴".

David's script and narration describes counting people one at a time and then two at a time, introducing variables, loops, and conditions.

³ <http://guide.cs50.net>

⁴ <http://ed.ted.com/lessons/your-brain-can-solve-algorithms-david-j-malan>

Programming Concepts

- A variable is like a container (a glass bowl) implemented as bits on a hard drive or memory, that stores a particular value.

```
let N = 0
for each person in room
  set N = N + 1
```

In the code above, line 1 sets the value of `N` to 0, like having an empty glass bowl.
Line 3 is like adding a ping pong ball to the glass bowl, incrementing the value of the variable by 1.

- A loop is doing the same thing over and over again, commonly expressed in a format like "for each person in the room, do this" with the proposition `for`.

The indented line, line 3, is will be done in every pass of the loop.

- A condition, or branch, will handle certain cases. In the animated example, when counting pairs of people in a room, the case of having an odd number of people is handled by such a condition, as shown below.

```
if 1 person remains then
  set N = N + 1
```

PB&J Demonstration

- Rob and two volunteers, Anita and Kiersten, will demonstrate making a peanut butter and jelly sandwich. David will note pseudocode instructions from the audience.

Brief clip of [It's Peanut Butter Jelly Time!!!](#)⁵ from the internet.

Rob will wear Google Glass to record his point of view.

- Computers need very specific instructions to complete tasks, and here audience volunteers will give instructions.

1. Open the bag of bread.
2. Remove the bread.

⁵ <https://www.youtube.com/watch?v=s8MDNFaGfT4>

Rob throws the bag of bread away from the table.

3. Place two slices next to each other.

4. Take your hand and set it lightly atop the peanut butter and unscrew the lid and put it gently next to the peanut butter.

5. Pick up knife.

Rob holds knife by blade.

6. Hold knife by the handle.

7. Put knife in peanut butter and take out as little as possible.

Rob cuts through the paper seal of the peanut butter.

8. Btw, remove paper first.

9. Using knife in peanut butter, apply peanut butter on said bread.

10. Taste peanut butter to assess quality.

11. Carefully pick up jelly.

Rob holds bottle upside-down.

12. Remove lid from jelly and barely squeeze any.

13. Btw, remove paper from jelly.

14. Invert jelly bottle before jelly falls out.

15. Replace the cap.

16. Remove cap from jelly.

17. Remove cap from peanut butter.

18. Drop all the jelly on the bread.

19. Remove excess jelly.

20. While any sandwich remains

21. Eat

- This demonstration (and Rob in particular) reveals how humans take for granted that others will understand what they mean, but computers need explicit, literal instructions.
- Ultimately, we will program in source code that is not in English but a language like C or Java or C++.

- The following code looks scary, but will be understandable by next week.

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

- The syntactic stuff, braces and semicolons, is not as important as the "simple ideas." In this case, we want to focus on the line that begins with `printf` and simply has the effect of saying `hello, world`.

Scratch

- To work with just the important ideas for now, we will start with a programming language called [Scratch](http://scratch.mit.edu)⁶ from MIT for Problem Set 0. The Scratch software can be used in the browser, or downloaded for use offline. The software allows us to write programs that use graphical puzzle pieces to instruct programs using the same concepts of variables, loops, and conditions. Its mascot, a cat, is also named Scratch.
- The `Control` category contains blocks with familiar words like `if`, `else`, and `repeat`.
The `[when (green flag) clicked]` block is what starts the program.
- The `Looks` category has a block named `[say []]` that has a box for us to type text into.
By dragging that block below the `[when (green flag) clicked]` block, the cat will say the text, in this case, `hello, world`, when the green flag is clicked.
- We can call each instruction a **statement**. Programs are generally many algorithms that humans wrote, in essence zeroes and ones, that represent statements that together solve a problem.
Examples of statements are the `wait` block that pauses the program and the `play sound` block that plays sounds.
- A **boolean expression** is either `true` or `false`.

⁶ <http://scratch.mit.edu>

`<mouse down?>`, `< [x] < [y] >`, `<touching [mouse-pointer]>` are all boolean expressions.

The `< < >` and `< > >` block can be used to combine boolean expressions.

- A **condition** such as the `[if <>]` block will have hexagonal openings that fit boolean expression blocks, which are in the shape of a hexagon.

Conditions can also be nested for three possible branches.

- **Loops** include the `forever` and `repeat` blocks.
- **Variables** are represented by blocks such as `[set [N] to [0]]` where `N` and `0` can be arbitrarily defined.
- Demonstrations of [Pastry Catch](#)⁷ with Abby and [Ivy's Hardest Game](#)⁸ with Phillip.
- Let's build a sample Scratch program using all of these concepts.

First, add an `[if < > then]` block to the `when green flag clicked` block.

Then add `[pick random [1] to [10]]` to the left side and `8` to the right side, and `[play sound [meow]]` to the inside of the condition, resulting in an 80% chance of Scratch saying "meow."

[Starting over](#)⁹, we can use a `[forever]` block and a `[wait [1] second]` block to have Scratch say "meow" every second.

- In [pet the cat](#)¹⁰, we combine a loop and a condition so that the cat will meow only if the mouse pointer is touching it or, in other words, if we are petting it. In [don't pet the cat](#)¹¹, we add an extra condition (using the `else` keyword) so that the cat will meow indefinitely, but will roar if we touch it with the mouse pointer.
- [hi hi hi](#)¹² contains a seal barking, but will stop when the space bar is pressed. One block controls the barking of the seal and one waits for space to be pressed, but both loops run simultaneously.

⁷ <http://scratch.mit.edu/projects/2016536/>

⁸ <http://scratch.mit.edu/projects/12352154/>

⁹ <http://scratch.mit.edu/projects/12199099/>

¹⁰ <http://scratch.mit.edu/projects/12199100/>

¹¹ <http://scratch.mit.edu/projects/12199106/>

¹² <http://scratch.mit.edu/projects/12199112/>

- [counting sheep](#)¹³ contains a sheep that says "baa" with some probability.
- The seal and sheep are called **sprites**, each with a different **costume**. Each sprite can have its own set of blocks controlling its actions, and move or act simultaneously, which is what we call **threading**.
- Another program written by Dan Bradley has a Scratch **extension**, or custom puzzle piece, that controls the lightbulb on the lectern. It sends a message across the internet to do this.
 - # He has also implemented buttons that change the lightbulb to various colors, as well as controlling the entire binary bulbs fixture, through Scratch.

Announcements

- Sections and office hours will begin next week.
- Zamyra Chan's walkthroughs will help you begin problem sets.
- Finally, we play [What Does the Fox Say](#)¹⁴ as a final example of Scratch.

¹³ <http://scratch.mit.edu/projects/12198996>

¹⁴ <http://scratch.mit.edu/projects/26329254/>